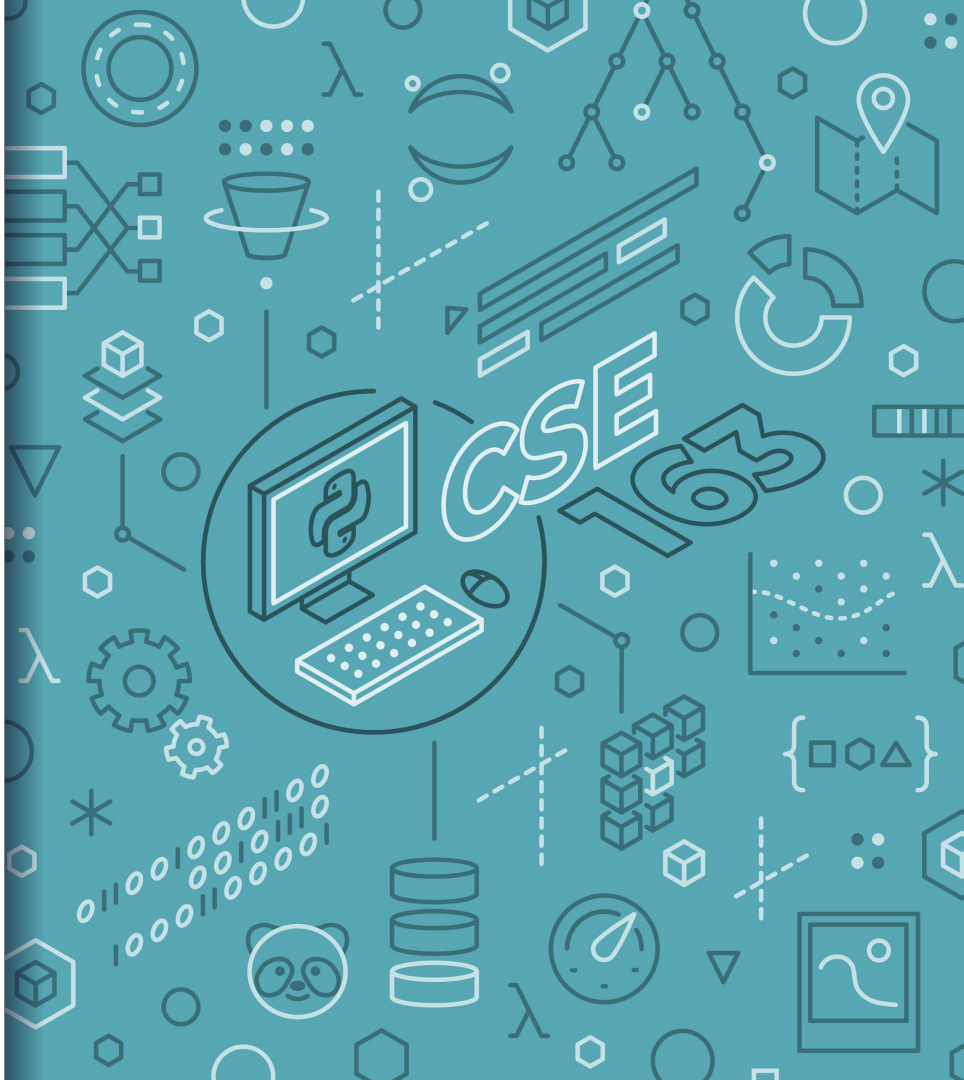
CSE 163

Modules, Packages and Processing Text Data

Hunter Schafer



Class

- A **class** lets you define a new object type by specifying what state and behaviors it has
- A class is a blueprint that we use to construct **instances** of the object

Here is a full class

```
class Dog:
  def __init__(self, name):
    self.name = name

  def bark(self):
    print(self.name + ': Woof')
```

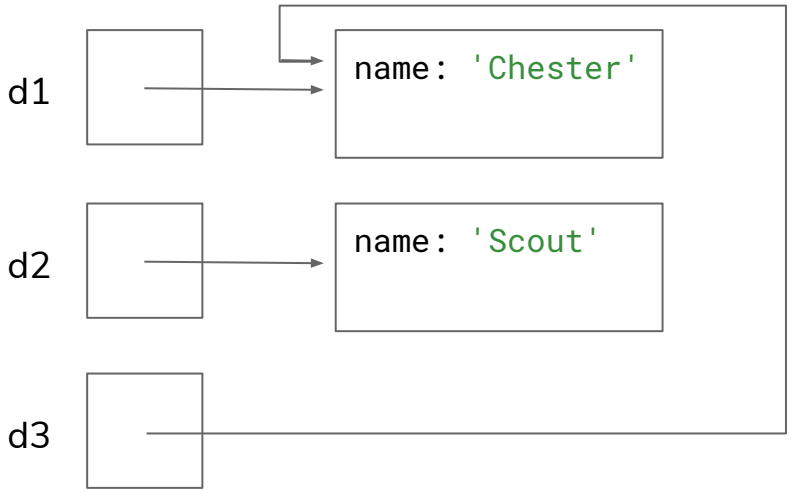
A class definition

An initializer that sets fields **(state)**

A method **(behavior)**

Building Dogs

```
d1 = Dog('Chester')
d2 = Dog('Scout')
d3 = d1
d1.bark()  # Chester: Woof
d2.bark()  # Scout: Woof
d3.bark()  # Chester: Woof
```



Poll Everywhere

Pair

2 minutes

pollev.com/cse163

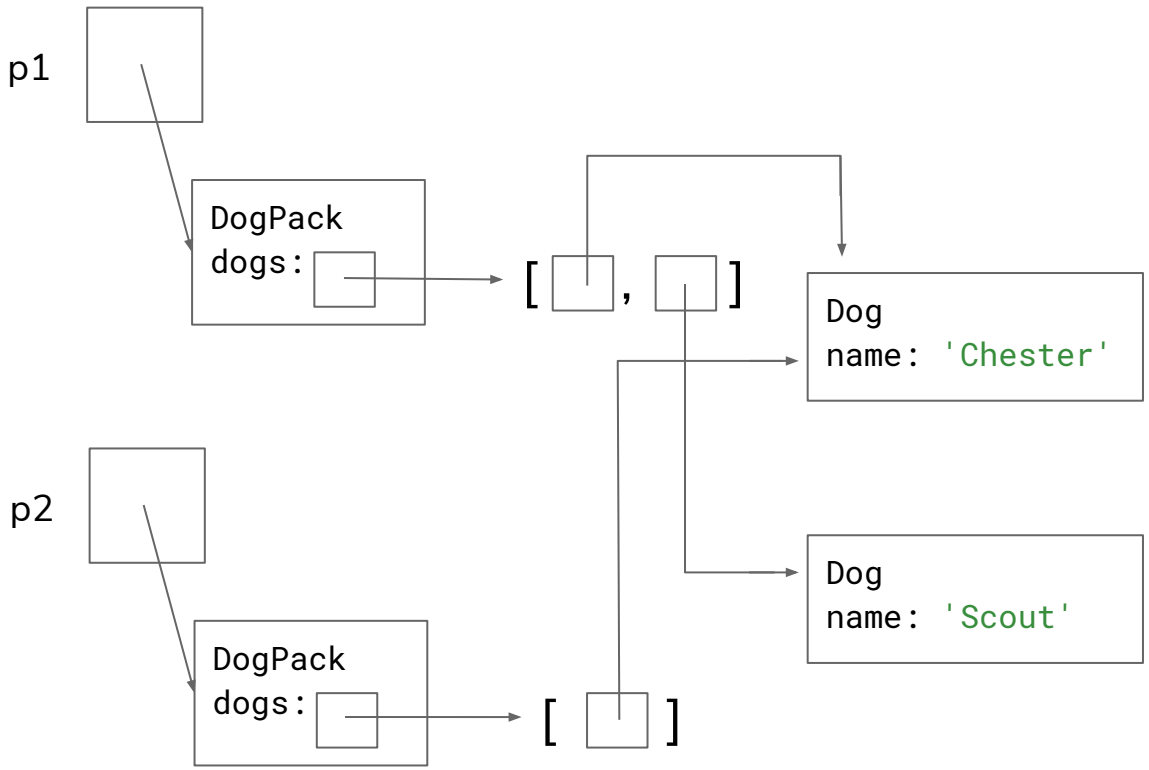
For this program, draw the memory model for the objects and then select which option best represents your model.

```
d1 = Dog('Chester')
d2 = Dog('Scout')
d3 = d1

p1 = DogPack()
p1.add_dog(d1)
p1.add_dog(d2)

p2 = DogPack()
p2.add_dog(d3)
```

Poll Everywhere

Think

pollev.com/cse163



Private

- Python has no way to actually do this, but by convention people don't access things that start with "_"

```
class DogPack:
    def __init__(self):
        self._dogs = []

    def add_dog(self, dog):
        self._dogs.append(dog)

    def _private_method(self):
        print('Some helper method')
```

Main Method Pattern

- Why have we been making you do this annoying pattern?

```
def main():
    print('Hello world')

if __name__ == '__main__':
    main()
```

- If you don't, it will run the main method if you import the file!
  - Usually not fun to run a 2 hour data analysis if you just wanted to import one helper function.

VS Code

Default Parameters

VS Code

- You can use default parameters like you would before
- You have to be careful when using objects as default values, it has some really bad unintentional side-effects

```
def fun(param=[]):
    param.append(1)
    print(param)

fun([2])  # [2, 1]
fun([2])  # [2, 1]
fun()     # [1]
fun()     # [1, 1]
```

- There is only one instance of the default parameter, they share a reference!

Default Parameters Done Right

VS Code

- The fix is to not use an object as the default parameter, instead we usually use None

```
def fun(param=None):
    if param is None:
        param = []
    param.append(1)
    print(param)

fun([2])  # [2, 1]
fun([2])  # [2, 1]
fun()     # [1]
fun()     # [1]
```

How to Run a Python Program

- Python looks relative to where you are running the program

```
dir
├── dogs.py
└── main.py
```

- Inside main.py, I could import

```
import dogs
```

- If I'm inside dir, I can run things and it will look relative to where I am running the Python program

```
(/path/to/dir)$ python main.py
```

Code Organization

- A **module** corresponds to a single Python file
  - It can have functions, classes, and statements inside of it
- A **package** corresponds to a group of Python files (folder)
  - It can contain modules or other packages

How to Run a Python Program

- What if I wanted to make a pets package

```
dir
├── main.py
└── pets
    └── dogs.py
```

- We have to change how we import it now

```
~~import dogs~~
import pets.dogs
```

- Running the program is the same though

```
(/path/to/dir)$ python main.py
```

__init__.py

- By default, Python usually doesn't convert folders to packages
- You need to put a special file named __init__.py in the folder to make it a package
- This file can be empty

```
dir
├── main.py
└── pets
    ├── __init__.py
    └── dogs.py
```

- You don't need one in your top-level directory, because you are not going to import it!

Brain Break

Unstructured Text

- So far we have seen “structured” text. Has nice orderly format.
- Most text in the world is unstructured (free-form)
  - Books, Wikipedia, Tweets
- The techniques we need to process this data are pretty different
- Today’s data: Wikipedia articles about people

Representation Matters

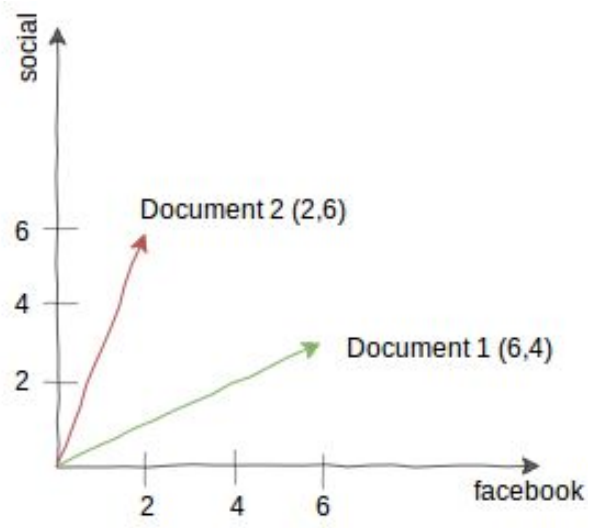
- When working with natural language, we use different representations of the data to compute different properties
- In this class, we will see
  - Bag of Words
  - TF-IDF
  - Unigram
  - Bi-gram
- Today we are just talking about Bag of Words

- Goal: Identify similarity between two documents

Bag of Words

- Very simple representation
  - Make a dictionary that maps words to their counts
- We use this to think of each document like a vector
  - Really just a point in space
- There is one dimension for each word. This is HUGE.

	doc1	doc2
facebook	6	2
social	3	6



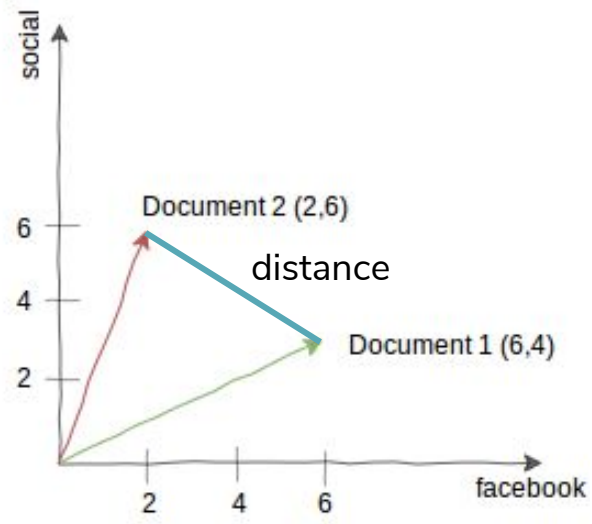
Dimensions

- Technically, we would need to have an entry for every possible word in the language so we could compare the vectors.
- We will use a **sparse** representation
  - Each document will only store keys that it has counts for
  - Any missing keys are considered 0.
- This saves a lot of space for each document, but the dimensionality of these things is still really big.

Distance

- Now that each document is a vector that lives in a space, we can actually talk about the “distance” between one document and another
- The “distance” between 2 documents is the distance between their vectors

	doc1	doc2
facebook	6	2
social	3	6



Euclidean Distance

- You may have seen this before in a math class
- The euclidean distance between two vectors **p** and **q**

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

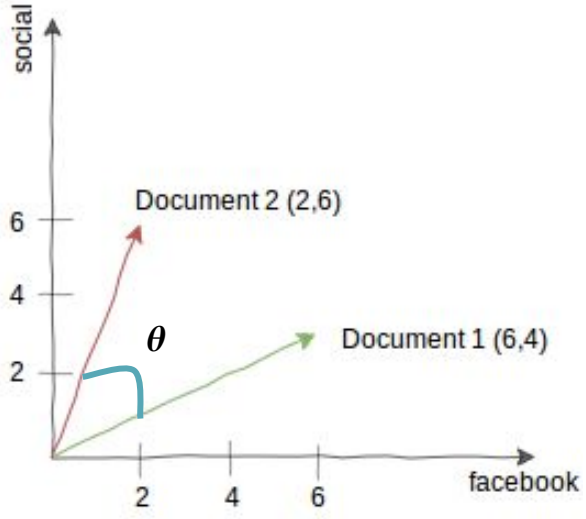
- This is just a random formula from linear algebra, don't memorize it.

VS Code

Euclidean Distance + NLP

- Euclidean distance does not really work with these bag of words
- It cares too much about the magnitudes of the vectors
  - This causes weird effects based on the length of the documents
- What we care about is that the documents use similar sets of words rather than penalizing differences in counts

	doc1	doc2
facebook	6	2
social	3	6



Cosine Distance

- Instead, measure the angle between the vectors
- Use cosine-similarity find how similar they are
  - Just a random formula from linear algebra, don't memorize this

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

- This is a measure of similarity, so to get distance we take

  1 - similarity